AMENDMENTS TO THE CLAIMS

Please amend the claims as follows.

1. - 38 (Cancelled)

39. (New) A method of controlling a computer system comprising:

establishing a privileged region of memory for executing code in privileged mode;

establishing a non-privileged region of memory for executing code in non-privileged mode;

receiving a memory access request to access a memory address;

determining whether the memory address is in the privileged region of memory in response to the request; and

switching the system to privileged mode if the memory address is determined to be in the privileged region of memory.

40. (New) The method of claim 39, further comprising:

switching the system to non-privileged mode if the memory address is determined not to be in the privileged region of memory.

41. (New) The method of claim 39, wherein determining whether the memory address is in the privileged region of memory is performed during a translation of a virtual address to a physical address.

42. (New) The method of claim 39, wherein the memory access request is a system call, and wherein the system call is implemented as a standard function call.

43. (New) The method of claim 42, further comprising:

switching the system to non-privileged mode at the conclusion of the system call.

44. (New) The method of claim 42, wherein switching the system to privileged mode further comprises switching the system to privileged mode if the system call is not initiated from the privileged region of memory.

45. (New) The method of claim 39, wherein determining whether the memory address is in the privileged region of memory comprises comparing the address against predetermined address limits.

46. (New) The method of claim 39, wherein the memory is divided into a plurality of pages, and wherein determining whether the memory address is in the privileged region of memory comprises:

identifying a page of the plurality of pages, wherein the page comprises the memory address; and

determining whether an indicator associated with the page identifies the page as a page in the privileged region of memory.

47. (New) The method of claim 46, wherein the indicator is stored in a page translation table.

48. (New) The method of claim 46, wherein a first value of the indicator identifies the page as in the non-privileged region and accessible from the privileged region and the non-privileged region, wherein a second value of the indicator identifies the page as in the privileged region and accessible from the privileged region and the non-privileged region, and wherein a third value of the indicator identifies the page as in the privileged region and accessible only from the privileged region.

49. (New) The method of claim 39, wherein the privileged region is divided into a first privileged sub-region and a second privileged sub-region, and wherein a function call from the non-privileged region is permitted into only the first privileged sub-region.

50. (New) The method of claim 39, wherein the privileged region and the non-privileged region are established during system initialization.

51. (New) The method of claim 39, wherein at least a portion of device driver code is located in the privileged region.

52. (New) The method of claim 39, wherein at least a portion of trusted application code is located in the privileged region.

53. (New) A computer system comprising:

means for establishing a privileged region of memory for executing code in privileged mode;

means for establishing a non-privileged region of memory for executing code in non-privileged mode;

means for receiving a memory access request to access a memory address;

means for determining whether the memory address is in the privileged region of memory in response to the request;

means for switching the system to privileged mode if the memory address is determined to be in the privileged region of memory.

54. (New) The computer system of claim 53, further comprising:

means for switching the system to non-privileged mode if the memory address is determined not to be in the privileged region of memory.

55. (New) A computer system comprising:

a privileged region of memory for executing code in privileged mode; and

a non-privileged region of memory for executing code in non-privileged mode,

wherein the computer system receives a memory access request to access a memory address, determines whether the memory address is in the privileged region of memory in response to the request, and switches the system to privileged mode if the memory address is determined to be in the privileged region of memory.

56. (New) The computer system of claim 55, wherein the computer system further switches the system to non-privileged mode if the memory address is determined not to be in the privileged region of memory.

57. (New) The computer system of claim 55, wherein the computer system determines whether the memory address is in the privileged region of memory during a translation of a virtual address to a physical address.

58. (New) The computer system of claim 55, wherein the memory access request is a system call, and wherein the system call is implemented as a standard function call.

59. (New) The computer system of claim 55, wherein the memory is divided into a plurality of pages, and wherein the computer system determines whether the memory address is in the privileged region of memory by:

identifying a page of the plurality of pages, wherein the page comprises the memory address; and

determining whether an indicator associated with the page identifies the page as a page in the privileged region of memory.

60. (New) The computer system of claim 59, wherein a first value of the indicator identifies the page as in the non-privileged region and accessible from the privileged region and the non-privileged region, wherein a second value of the indicator identifies the page as in the privileged region and accessible from the privileged region and the non-privileged region, and wherein a third value of the indicator identifies the page as in the privileged region and accessible only from the privileged region.

61. (New) The computer system of claim 59, wherein the indicator is stored in a page translation table.

62. (New) A computer program product comprising program instructions embodied thereon for causing a computer system to:

establish a privileged region of memory for executing code in privileged mode;

establish a non-privileged region of memory for executing code in non-privileged mode;

receive a memory access request to access a memory address;

determine whether the memory address is in the privileged region of memory in response to the request;

switch the system to privileged mode if the memory address is determined to be in the privileged region of memory.

63. (New) The computer program product of claim 62, wherein the program instructions further cause the computer system to switch the system to non-privileged mode if the memory address is determined not to be in the privileged region of memory.

64. (New) The computer program product of claim 62, wherein the program instructions further cause the computer system to determine whether the memory address is in the privileged region of memory during a translation of a virtual address to a physical address.

65. (New) The computer program product of claim 62, wherein the memory access request is a system call, and wherein the system call is implemented as a standard function call.

66. (New) The computer program product of claim 65, wherein the program instructions further cause the computer system to:

switch the system to a non-privileged mode at the conclusion of the system call.

67. (New) The computer program product of claim 65, wherein the program instructions further cause the computer system to switch the system to privileged mode if the system call is not initiated from the privileged region of memory.

68. (New) The computer program product of claim 62, wherein the program instructions further cause the computer system to determine whether the memory address is in the privileged region of memory by comparing the memory address against predetermined address limits.

69. (New) The computer program product of claim 62, wherein the memory is divided into a plurality of pages, and where the program instruction further cause the computer system to determine whether the memory address is in the privileged region of memory by:

identifying a page of the plurality of pages, wherein the page comprises the memory address; and

determining whether an indicator associated with the page identifies the page as a page in the privileged region of memory

70. (New) The computer program product of claim 69, wherein the indicator is stored in a page translation table.

71. (New) The computer program product of claim 69, wherein a first value of the indicator identifies the page as in the non-privileged region and accessible from the privileged region and the non-privileged region, wherein a second value of the indicator identifies the page as in the privileged region and accessible from the privileged region and the non-privileged region, and wherein a third value of the indicator identifies the page as in the privileged region and accessible only from the privileged region.

72. (New) The computer program product of claim 62, wherein the privileged region is divided into a first privileged sub-region and a second privileged sub-region, and wherein a function call from the non-privileged region is permitted into only the first privileged sub-region.

73. (New) The computer program product of claim 62, wherein the privileged region and the non-privileged region are established during system initialization.

74. (New) The computer program product of claim 62, wherein at least a portion of device driver code is located in the privileged region.

75. (New) The computer program product of claim 62, wherein at least a portion of trusted application code is located in the privileged region.